

쾌적한 Clone Checker

OOPT 2nd Cycle – System Test

Software Modeling & Analysis

유준범 교수님

Team. T1

201111388 조연호

201211374 이창오

201211379 장중훈

201314196 양동혁

목차

Contents

- ✓ Revise Plan
- ✓ Test Report
- ✓ Test Respond Report
- ✓ Demonstration

Revise Plan

Stage 1000, Stage 2030, Stage 2040, Stage 2050 보고서 내
모호한 내용 명확하게 명세 / 불필요한 내용 삭제

```

Files
- name: String
- numOfLine: int
- numofFunction: int
- numofVariable: int
- numofPreprocessor: int
- numofAnnotation: int
- listFunction: ArrayList<String>
- listVariable: ArrayList<String>
- listPreprocessor: ArrayList<String>

+ Files(fileName)
+ getName(): String
+ getNumOfLine(): int
+ getNumOfFunction(): int
+ getNumOfVariable(): int
+ getNumOfPreprocessor(): int
+ getNumOfAnnotation(): int
+ getListFunction(): ArrayList<String>
+ getListVariable(): ArrayList<String>
+ getListPreprocessor(): ArrayList<String>
+ setName(tempName): void
+ setNumOfLine(tempNumOfLine): void
+ setNumOfFunction(tempNumOfFunction): void
+ setNumOfVariable(tempNumOfVariable): void
+ setNumOfPreprocessor(tempNumOfPreprocessor): void
+ setNumOfAnnotation(tempNumOfAnnotation): void
+ setListFunction(tempFunctionName): void
+ setListVariable(tempVariableName): void
+ setListPreprocessor(tempPreprocessorName): void

```

```

Controller
- folderPath: String
- fileName: String
- files: ArrayList<Files>
+ source: ArrayList<String>
- pointerNum: int
- fileNum: int
- numOfFile: int

+ Controller()
+ mkFileInstance(): void
+ displayMain(): void
+ selectFolder(): string
+ showFolderPath(): void
+ start(): void
+ displayResult(): void
+ showCloud(): void
+ displaySync(pointerNum, cal, lbTempSync, lbTempTotalSyncRate, lbTempSyncRate): void
+ changeCenter(tempPointerNum): void
+ showDialog(): int
+ getFolderPath(): String

```

```

Analyze
+ source: ArrayList<String>
+ tempFolderPath: String
+ numOfLine: int
+ numofFunction: int
+ numofVariable: int
+ numofPreprocessor: int
+ numofAnnotation: int
+ listFunction: ArrayList<String>
+ listVariable: ArrayList<String>
+ listPreprocessor: ArrayList<String>
+ listType: ArrayList<String>

+ Analyze(source)
+ analyzeFile(file): void
+ changeAnnotation(): void
+ analyzeAnnotation(): void
+ analyzeLine(): void
+ deleteAnnotation(): void
+ divideCodeLine(): void
+ deletePrintf(): void
+ findType(): void
+ analyzeFunction(): void
+ analyzeVariable(): void
+ analyzePreprocessor(): void

```

```

Calculate
- syncRate: float[][]
- centerNum: int
- fileNum: int
- numOfFile: int

+ Calculate(files)
+ calFile(): void
+ calLine(): void
+ calFunction(): void
+ calVariable(): void
+ calPreprocessor(): void
+ calAnnotation(): void
+ setNumCenter(pointerNum): void
+ getTotalSync(pointerNum): float
+ getLineSync(pointerNum): float
+ getFunctionSync(pointerNum): float
+ getVariableSync(pointerNum): float
+ getPreprocessorSync(pointerNum): float
+ getAnnotationSync(pointerNum): float

```

divideCodeLine()
소스 코드를 여러 줄로
잘못 나누는 오류 수정

findType()
소스 코드 내 새로 선언된
타입을 찾는 메소드 추가

mkFileInstance()
분석이 불가능한 폴더 선택 시
예외를 발생시키도록 수정

calLine(),
calFunction(),
calVariable(),
calPreprocessor(),
calAnnotation()
일치율 계산 알고리즘
변경으로 인한 정확성 증가

Test Report

1조

- ✓ Specification Review
23개 발견
- ✓ Brute Force Testing
총 61개 중 1개 성공
- ✓ Category Partition Testing
총 128개 중 0개 성공
- ✓ Pairwise Combination Testing
총 128개 중 0개 성공

5조

- ✓ Specification Review
14개 발견
- ✓ Brute Force Testing
4개 실패
- ✓ Category Partition Testing
총 64개 중 47개 성공
- ✓ Pairwise Testing
총 8개 중 6개 성공

Test Respond Report _ Specification, 1조

위치	Stage 1000 - Activity 1001. Define Draft Plan
문제	기본 문서에서 프로그램의 Object로 쉽게 보여주는 Clone Checker를 지향하였지만 쉽게 보이지 않았다.
대응	결과 화면 창에 출력되는 클라우드 내 텍스트와 리스트 내 텍스트의 출력 형태와 크기를 변경하였다.

위치	Stage 1000 - Activity 1003. Define Requirements
문제	Activity 1003(Define Requirements)의 Functional Requirements 3.7과 3.8의 설명이 같고 그 설명마저도 모호하다.
대응	R.3.7(Analyze Function)의 설명을 '소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.'로 수정하였다. R.3.8(Analyze Variable)의 설명을 '소스 코드 파일의 변수의 개수와 이름을 분석하여 결과값을 저장한다.'로 수정하였다.

위치	Stage 2030 - Activity 2131. Define Essential Use Cases Stage 1000 - Activity 1007. Define Business Use Case
문제	OSP 2030 Stage의 Activity 2131(Define Essential Use Cases)에서 Divide Code Line와 관련된 Use Case의 코드 분할기준이 OSP 1000 Stage의 Activity 1007(Define Business Use Case)의 Describe Use Cases에서 Divide Code Line use case의 설명에 비해 다소 추상적이다.
대응	Stage 1000 - Activity 1007. Define Business Use Case의 Description과 Stage 2030 - Activity 2131. Define Essential Use Cases의 Purpose의 내용을 '소스 코드 파일에서 한 줄에 여러 개의 코드가 있는 경우 ','을 기준으로 행을 나눈다.'로 구체화하여 동기화하였다.

Test Respond Report _ Specification, 1조

위치	Stage 2040 - Activity 2141. Design Real Use Cases
문제	OSP 2040 Stage의 Use Case “Select folder”에서 Pre Requisites의 내용과 Typical Courses of Events의 내용이 겹치고 Exception처리가 반영되어 있어야 하지만 문서상에서는 발견되지 않았다.
대응	Typical Courses of Events에서 Pre Requisites와 중복되었던 ‘버튼(A)를 누른다.’ 내용을 삭제하였다. Exceptional Courses of Events를 제거하였다.

위치	Stage 2040 - Activity 2141. Design Real Use Cases
문제	OSP 2040 Stage의 Use Case “Analyze Line”에서 Typical Courses of Events의 1번에서 개수 파악이 어느 개수인지 모르겠다.
대응	Typical Courses of Events의 내용을 아래와 같이 구체화하였다. - (A) : Actor, (S) : System 1. (S) : 생성된 Analyze:source의 문장 개수를 파악하여 소스 코드 파일의 라인 수를 분석한다. 2. (S) : 라인 수를 File:numOfLine에 저장한다.

Test Respond Report _ Specification, 1조

위치	Stage 2040 - Activity 2141. Design Real Use Cases
문제	OSP 2040 Stage의 Use Case “Calculate Preprocessor Sync-Rate”에서 Typical Courses of Events의 내용 중 “오차범위(30,10)”의 내용이 정확히 무엇을 의미하는지 문서의 내용을 읽어만 봐서는 알기 힘들다.
대응	<p>Calculate Preprocessor Sync-Rate의 Typical Courses of Events을 아래와 같이 구체화하였다.</p> <p>-</p> <p>(A) : Actor, (S) : System</p> <p>1.(S) : 기준 파일의 전처리기 수가 비교 파일의 전처리기 수보다 클 경우 ‘{(비교 파일과 기준 파일에서 이름이 같은 전처리기 수 / 기준 파일의 전처리기 수) * 100} * 0.6 + {(비교 파일의 전처리기 수 / 기준 파일의 전처리기 수) * 100} * 0.4’ 수식의 결과값을 일치율로 부여한다.</p> <p>2.(S) : 일치율을 Calculate:syncRate에 저장한다.</p> <p>-</p> <p>Calculate Preprocessor Sync-Rate의 Alternative Courses of Events을 아래와 같이 구체화하였다.</p> <p>-</p> <p>(A) : Actor, (S) : System</p> <p>1. (S) : 기준 파일의 전처리기 수가 비교 파일의 전처리기 수보다 작을 경우 ‘{(비교 파일과 기준 파일에서 이름이 같은 전처리기 수 / 비교 파일의 전처리기 수) * 100} * 0.6 + {(기준 파일의 전처리기 수 / 비교 파일의 전처리기 수) * 100} * 0.4’ 수식의 결과값을 일치율로 부여한다.</p> <p>2. (S) : 일치율을 Calculate:syncRate에 저장한다.</p>

Test Respond Report _ Specification, 5조

위치	Stage 1000 - Activity 1003. Define Requirements
문제	요구 사항 항목 “R.1, Select Folder, 비교할 소스 코드 파일들이 들어있는 폴더를 선택한다.” 에서 “소스 코드” 와 “소스 코드 파일”에 대한 정의가 불분명하다. 소스 코드는 어떤 프로그래밍 언어로 작성되어야 하는가? 해당 프로그래밍 언어의 명세 상으로 유효한, 즉 컴파일 가능한 코드여야 하는가? 소스 코드 파일의 크기에는 어떤 제한이 있는가?
대응	Stage 1000 - Activity 1004. Record Terms in Glossary에 다음과 같은 내용을 추가하였다. - 소스 코드 : 컴파일 가능 여부와 상관 없이 프로그래밍 언어로 작성된 코드 소스 코드 파일 파일 크기와 상관 없이 소스 코드가 담긴 .c 확장자 파일

위치	Stage 1000 - Activity 1003. Define Requirements
문제	“B.4, 종료 버튼, 화면 하단에 위치하여 클릭 시 프로그램이 종료된다”, “C. 그 외의 메시지 출력은 대화상자 (Dialog) 활용” 이라고 명세되어 있는데, 종료 버튼과 대화 상자가 없다.
대응	종료 버튼에 관한 내용을 삭제하였고, 메시지 출력은 대화상자(Dialog)를 활용하도록 수정하였다. Other Requirements에 ‘새로운 폴더를 지정하기 위해서는 프로그램을 종료하고 다시 실행해야 한다.’는 내용을 추가하였다.

Test Respond Report _ Brute Force Testing, 1조

문제	ST-5. 개당 대용량 소스파일 비교 - Fail ST-6. 소스 파일 분석 - Fail ST-7. 파일 1개 분석 - Fail ST-8. 파일 2개 분석 - Fail ST-9. 파일 3개 분석 - Fail ST-10. 파일 4개 분석 - Fail ST-11. 파일 5개 분석 - Fail ST-12. 파일 6개~10개 분석 - Fail ST-13. 파일 11개~50개 분석 - Fail
원인	한 줄에 같이 작성된 코드들의 행을 나누는 divideCodeLine() 메소드 내에서 무한 루프가 발생되었다.
대응	divideCodeLine() 메소드의 알고리즘을 수정하였다.
문제	ST-17. 0바이트 파일 비교 - Fail
원인	사용자가 지정한 폴더에 있는 0바이트인 파일을 분석하는 오류가 발생하였다.
대응	폴더 내 0바이트 파일이 있는 경우 예외처리를 통해 오류 메시지를 출력하고 다시 시작 화면으로 돌아가도록 수정하였다.

Test Respond Report _ Brute Force Testing, 1조

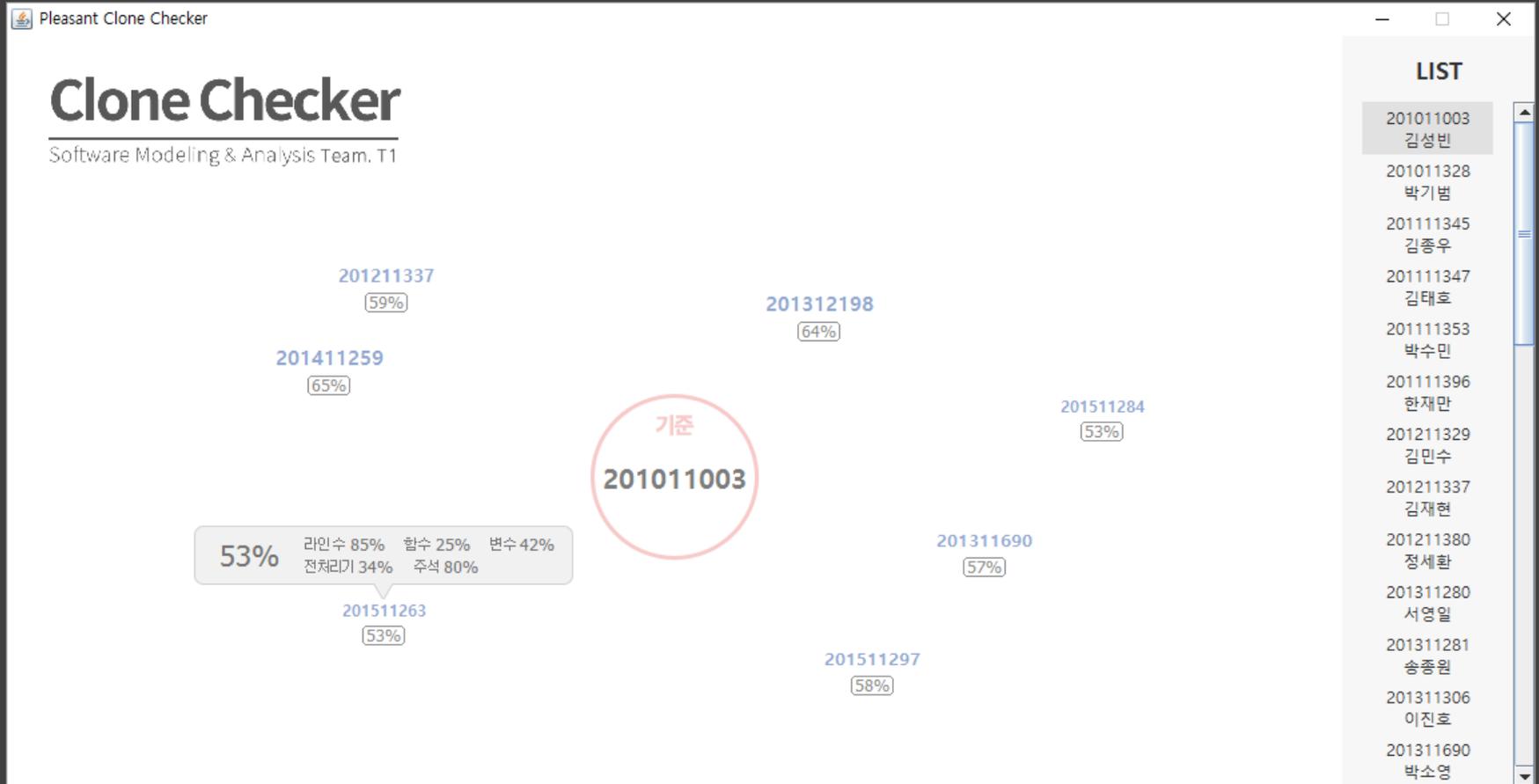
문제	ST-20. 같은 기준 선택시 변화 - Fail
원인	이미 기준 소스 코드 파일로 선택되어있는 파일로 기준 소스 코드 파일을 재선택할 경우 동일한 비교 작업이 수행되고, 클라우드 모양이 변경되었다.
대응	현재 기준 소스 코드 파일로 선택되어있는 파일로 재선택할 수 없도록 수정하였다.

문제	ST-24. 잘못된 폴더경로 참조 - Fail
원인	실재하지 않는 잘못된 폴더 경로를 지정하였을 경우에도 분석이 시작되는 오류가 발생하였다.
대응	실재하지 않는 잘못된 폴더 경로일 경우 예외처리를 통해 오류 메시지를 출력하고 다시 시작 화면으로 돌아가도록 수정하였다.

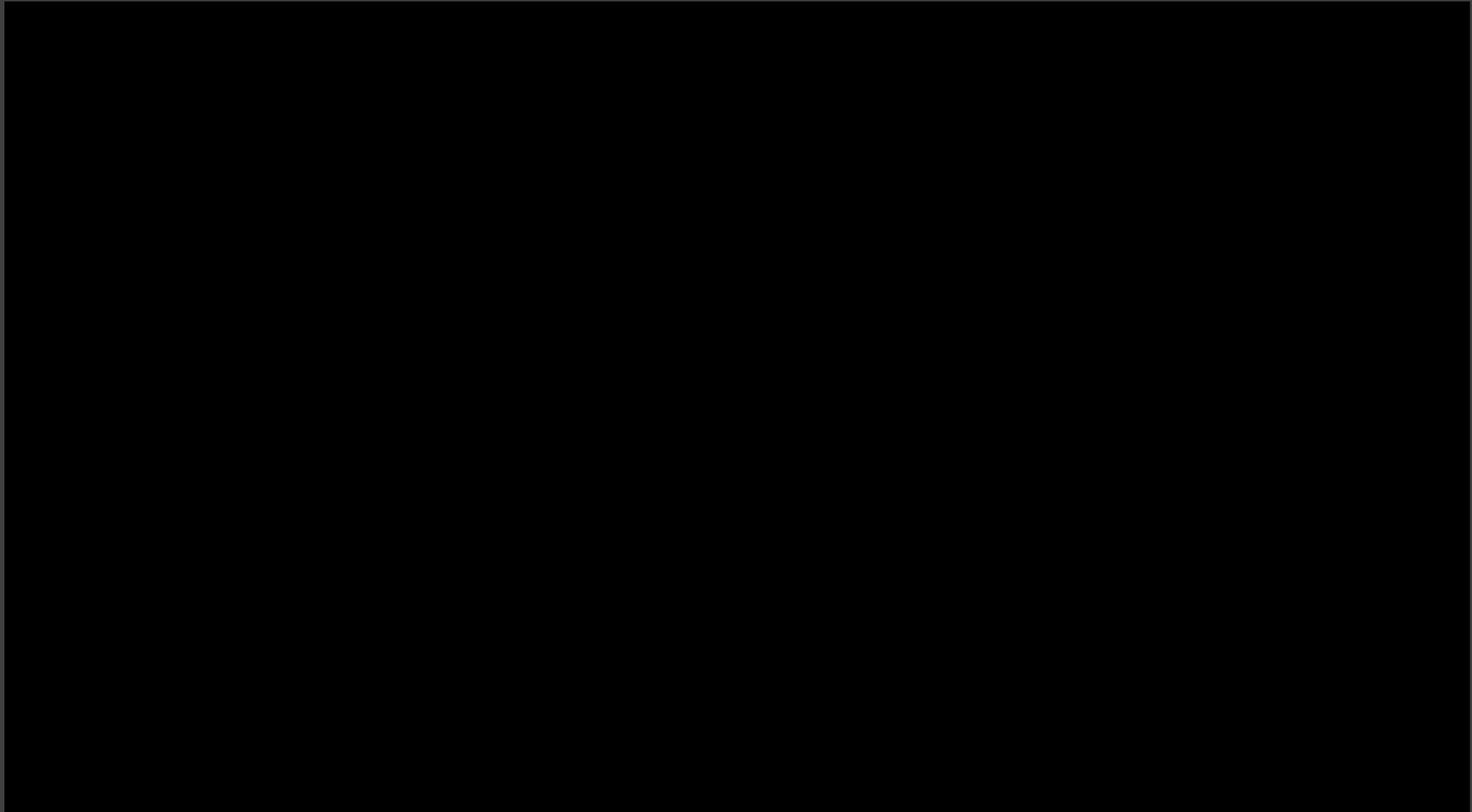
Test Respond Report _ Brute Force Testing, 5조

문제	세미콜론을 사용해 코드 여러 줄을 합쳤을 때 “Use Case: Divide Code Line”기능이 작동하지 않는다.
원인	한 줄에 같이 작성된 코드들의 행을 나누는 divideCodeLine() 메소드 내에서 무한 루프가 발생되었다.
대응	divideCodeLine() 메소드의 알고리즘을 수정하였다.

Test Respond Report _ GUI



Demonstration



<https://youtu.be/6asJ5fEBCX0>

— 쾌적한 Clone Checker

고맙습니다.